

1 What is claimed is:

2
3 1. A method of allocating registers when compiling source code, said method comprising
4 steps of:

5 translating source code to intermediate code;
6 identifying an operand from said intermediate code to store in a real register; and
7 selecting a class of real registers operable to store said operand.
8

9 2. The method of claim 1, further comprising steps of:
10 selecting at least one subclass of said selected class of real registers, wherein said at least
11 one subclass includes a register to store said operand.

12
13 3. The method of claim 1, wherein said selected class includes one of a callee-saved class
14 and a caller-saved class.
15

16 4. The method of claim 2, wherein said step of selecting at least one subclass further
17 comprises steps of:
18 selecting a first set of subclasses within said selected class;
19 determining whether a register included in said first set of subclasses is available to store
20 said operand; and
21 in response to said register being available, storing said operand in said register.
22

23 5. The method of claim 4, wherein said first set of subclasses includes at least one of non-
24 used-in-current-operation, non-busy, non-live and non-used subclasses.
25

26 6. The method of claim 4, wherein said step of selecting at least one subclass further
27 comprises steps of:
28 selecting a second set of subclasses within said selected class in response to said register
29 not being available in said first set of subclasses;
30 determining whether a register included in said second set of subclasses is available to
31 store said operand; and

1 in response to said register in said second set of subclasses being available, storing said
2 operand in said register in said second set of subclasses.

3
4 7. The method of claim 6, wherein said second set of subclasses includes at least one of
5 non-used-in-current-operation, non-busy, non-live and used subclasses.

6
7 8. The method of claim 6, wherein said step of selecting at least one subclass further
8 comprises steps of:

9 selecting a third set of subclasses within said selected class in response to a register in
10 said second set of subclasses not being available;

11 determining whether a register included in said third set of subclasses is available to store
12 said operand; and

13 in response to said register in said third set of subclasses being available, storing said
14 operand in said register in said third set of subclasses.

15
16 9. The method of claim 8, wherein said third set of subclasses includes at least one of non-
17 used-in-current-operation, live and non-busy subclasses.

18
19 10. The method of claim 8, wherein said step of selecting at least one subclass further
20 comprises steps of:

21 selecting a fourth set of subclasses within said selected class in response to a register in
22 said third set of subclasses not being available;

23 determining whether a register included in said fourth set of subclasses is available to
24 store said operand; and

25 in response to said register in said fourth set of subclasses being available, storing said
26 operand in said register in said fourth set of subclasses.

27
28 11. The method of claim 10, wherein said fourth set of subclasses includes at least one of
29 non-used in current operation and busy subclasses.

1 12. The method of claim 11, further comprising spilling a register in at least one of said busy
2 and said live subclasses prior to storing said operand in said register in at least one of said busy
3 and said live subclasses.

4
5 13. The method of claim 11, further comprising storing said operand in a class other than
6 selected class in response to a register in said fourth set of subclasses not being available.

7
8 14. The method of claim 11, further comprising marking said register as used-in-current-
9 operation in response to storing said operand in said register.

10
11 15. The method of claim 11, further comprising marking said register storing said operand as
12 live and not-used-in-current-operation in response to translating an instruction of said source
13 code.

14
15 16. The method of claim 1, further comprising steps of:
16 selecting another class of registers in response to said selected class of registers not
17 including a not used in current operation register; and
18 storing said operand in a register in said selected other class.

19
20 17. The method of claim 3, wherein said step of selecting a class further comprises steps of:
21 selecting said callee-saved class in response to said operand including at least one of local
22 variables, stack items and parameters input by a user; and
23 selecting said caller-saved class in response to said operand including a temporary
24 computation.

25
26 18. A method of compiling source code comprising steps of:
27 generating intermediate code from a portion of source code;
28 allocating a plurality of real registers to store a plurality of operands from said
29 intermediate code while generating the intermediate code; and
30

1 generating machine-readable code from said intermediate code using said plurality of real
2 registers.

3
4 19. The method of claim 18, further comprising a plurality of types of operands and said step
5 of allocating further comprises steps of:

6 determining a type of operand for at least one of said plurality of operands;
7 storing said at least one operand in memory in response to said operand being a particular
8 type of operand; and
9 allocating a real register for said operand.

10
11 20. The method of claim 19, wherein said particular type of operand includes a local variable.

12
13 21. The method of claim 19, wherein said step of allocating further comprises steps of:

14 selecting a class of registers depending on said type of operand; and
15 allocating a real register from said selected class of registers depending on said type of
16 operand.

17
18 22. The method of claim 21, wherein said step of selecting a class further comprises steps of:

19 selecting a first class of registers in response to said operand being at least one of a local
20 variable, a stack item and a parameter input by a user; and

21 selecting a second class of registers in response to said operand being a temporary
22 computation.

23
24 23. The method of claim 21, wherein said step of selecting allocating further comprises
25 selecting at least one subclass of registers in said selected class.

26
27 24. The method of claim 23, wherein said at least one selected subclass includes at least one
28 of live registers, non-live registers, busy registers, non-busy registers, used registers, non-used
29 registers, and non-used in current operation registers.

1 25. A compiler configured to compile source code into machine-readable code, said compiler
2 comprising:

3 a register allocation stage configured to generate intermediate code from said source code
4 and configured to allocate a plurality of real registers to a plurality of operands from said
5 intermediate code;

6 an optimization stage configured to optimize said intermediate code; and

7 a final code stage configured to generate said machine-readable code from said
8 intermediate code using said plurality real registers.
9

10 26. The compiler of claim 25, wherein said register allocation stage is configured to
11 determine a type of operand for at least one of said plurality of operands, and store said at least
12 one operand in memory in response to said operand being a particular type of operand, and
13 allocate a real register for said operand.
14

15 27. The compiler of claim 26, wherein said particular type of operand includes a local
16 variable.
17

18 28. The compiler of claim 25, wherein said register allocation stage is further configured to
19 select a class of registers and allocate a real register from said selected class of registers for one
20 of said plurality of operands, said one operand being of a particular type of operand.
21

22 29. The compiler of claim 28, wherein said register allocation stage is further configured to
23 select a first class of registers in response to said operand being a type including at least one of a
24 local variable, a stack item and a parameter input by a user; and

25 select a second class of registers in response to said operand being a temporary
26 computation.
27

28 30. The compiler of claim 28, wherein said register allocation stage is further configured to
29 select at least one subclass of registers in said selected class.
30

1 31. The compiler of claim 30, wherein said at least one selected subclass includes at least one
2 of live registers, non-live registers, busy registers, non-busy registers, used registers, non-used
3 registers, and non-used in current operation registers.
4

0953020-1091
T06T0T 0202560